# Evaluation Results

This section presents the effectiveness of the VV Python library in improving healthcare data visualization for academic research. We compare its performance against traditional methods. Our focus is on two metrics: "Time-to-First-Chart Draft" and "Time-to-Final-Chart."

The data for this evaluation was collected from a project that involved producing visual representations from a set of 72 spreadsheets. These times were captured using Monday.com, following a well-established practice within the organization where the author works for project management, including time-tracking. Time measurements for VV were taken using Python's time library, by calculating the delta of time between the start and completion of relevant tasks.

## Time Decomposition

The total time to complete the project was decomposed into two main categories:

**Initial Setup Time**

* For a human analyst, this refers to the time spent on organizing the spreadsheets and preparing the necessary files for task completion.
* For the VV system, this means the time required for adequately setting up the software environment and data linkage.

**Time per Spreadsheet**

* This is the time taken to generate a chart from each individual spreadsheet.

## Time Metrics

In Table X we compare the efficiency of manual methods against the Visual Viper Python library specifically for the aforementioned project with 72 spreadsheets.

Table X: Time Metrics Comparing Manual Methods and VV Python Library for a Project with 72 Spreadsheets.

| **Metric** |  | **Manual**  **Methods** | **Visual**  **Viper** |
| --- | --- | --- | --- |
| **Time-to-First-Chart-Draft** | **Initial setup** | 0h30min | 2h00min |
| **Time per spreadsheet** | 5min | <10-3 |
| **Total time (72 spreadsheets)** | 6h30min | 2h00min |
| **Time-to-Final-Chart** | **Initial setup** | 0h30min | 2h00 |
| **Time per spreadsheet** | 12min | To Miro: ~ 4 sec  To GDrive: ~3 sec |
| **Total time (72 spreadsheets)** | 14h54min | 2h9min |

VV: Visual Viper Library; h: hour; min: minute; sec: second.

## Adjustment for Fatigue

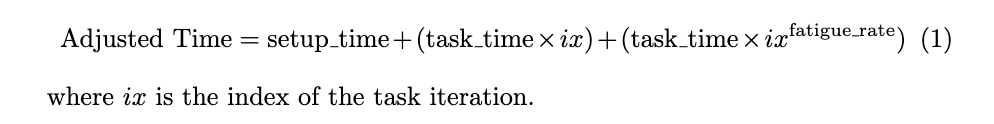
To enrich our evaluation, we extend the previous comparison by adding considerations for two essential factors. The analysis was performed using R (version 4.2.3) [[41]](https://paperpile.com/c/wYkhtl/4IRV) and the plots were generated using the ggplot2 package [[13]](https://paperpile.com/c/wYkhtl/1tUb).

We considered the following factors:

* **Task Fatigue**: It's acknowledged that task fatigue can affect the time taken for task completion in a non-linear manner..
* **Additional Human Intervention**: The output visualizations generated by VV requires additional human intervention for validation of accuracy, a factor not considered in the initial metrics.

In this simulation, we concentrate on the "Time-to-Final-Chart" metric, aiming to provide a more comprehensive view of the time required to produce a finalized chart, inclusive of all adjustments and confirmations.

The time adjusted for fatigue was computed using the equation (1):



We used bootstrapping with 100 samples, assuming a normal distribution for each variable. The 5th and 95th percentiles (P05 and P95) were calculated to construct 90% Confidence Intervals for our time metrics.

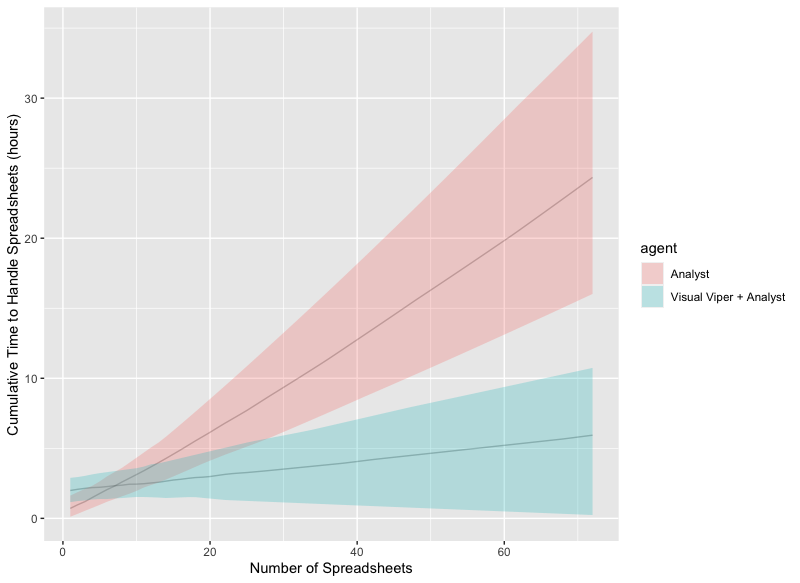


Figure X: Cumulative Time to Handle Spreadsheets for Different Agents

## Key Takeaways

The data presented in Table X and Figures X offer significant insights into the operational efficiencies associated with the VV Python library for chart creation in academic research. In particular, the differential impact of using VV in comparison to manual methods becomes more pronounced as the size of the project increases.

Figure X illustrates the cumulative time required to process 72 spreadsheets for both a standalone analyst and an augmented system involving both an analyst and VV. One of the striking observations is the crossover point where VV starts to show a time advantage. While the initial setup time for VV is significantly higher (2 hours compared to 0.5 hours for the analyst), the system starts to outperform the analyst alone at around 8 spreadsheets. By the time 25 spreadsheets are processed, the confidence intervals for the two methods no longer overlap, signaling a clear advantage for VV.

Our adjusted metrics also account for factors like task fatigue and the need for additional human verification of VV’s outputs. Even after these considerations, VV holds an advantage in larger projects, both in terms of time efficiency and likely in terms of reduced human error owing to fatigue.

Another significant aspect that adds complexity to this evaluation is the dynamic nature of these data collection processes. Studies are rarely static; they often require adjustments to the design or updating of data. These changes necessitate updating the charts, perhaps multiple times over the course of a study. While the initial setup is a one-time task, adjustments and updates are recurring tasks that continue to consume time. If the initial process is manual and lacks scalability, these frequent updates can quickly become a resource-consuming bottleneck. This is where the growing performance advantages of Visual Viper (VV) become particularly compelling. Our evaluation so far has considered only a single iteration of a project with 72 spreadsheets. In a dynamic study environment requiring frequent adjustments and updates, the scalability advantages of VV could be even more pronounced. Each update in a manual setting can be seen as an iteration that consumes substantial time and resources. VV, which already shows performance benefits in larger projects and single iterations, is likely to magnify these advantages in the context of ongoing, multiple iterations. Therefore, in a continually evolving study, the initial time investment in setting up VV is likely to yield significant long-term savings.